BuilConn™

From Web Server to Web Services

Concrete to Abstract
UI to M2M

OASIS

Web Servers are about UI

- Universal User Interface
- Develop Once, Run Anywhere
- Incremental Improvements to make more responsive
  - AJAX
- Still, it's all Eye-Candy
  - Can't feed your stock picker with two corporate web sites...

Web Services are not about UI

- Composite content
- Abstraction and Ontology
- Discovery Services

- SOAP (Simple Object Access Protocol)

Building Enterprise Protocols

BuilConn™

Challenges and issues facing business today

- Provide a flexible business model
  - The marketplace is changing - businesses need to change too
  - Many existing IT systems are inhibitors to change: complex and inflexible
  - Existing integrations can be inhibitors to change: multiple technologies, point-to-point integration, inflexible models
- Drive down cost
  - Eliminate duplicate systems
  - Re-use, don't re-build
  - Time to market
  - Simplify skills base
- Reduce cycle time and costs for external business processes
  - Move from manual transactions with suppliers towards automated transactions
  - Facilitate flexible dealings with partners with minimal process or IT impact
- Integrate across the enterprise
  - Integrate historically separate systems
  - Completion of mergers and acquisitions
  - Across physical and technology barriers

OASIS

SOAP

- “SOAP Lets computers surf the Web for data like people surf the Web for eye candy.”

  But how do computers know what they are looking at?

- Unstructured content → structured standards
- Concrete Content → Abstractions
- “Normal” language → Ontology

Phase 1: Anything Goes

- XML Tagging of Content
- Negotiation with Each Trading Partner
- Each XML document serves a single purpose
- [Expensive] Re-tooling with each change of partner
- Everything is possible because you can do what you want
- **Focus on Process**

Phase II: Standardization

- Adoption of standard data elements
  - EBXML
- Adoption of standard frameworks
  - WSRF, etc
- Still requires re-programming for each new purpose

Phase III: Composition

- WSDM (WS Distributed Management)
  - Esp. WSDM-MUWS
- BPEL (Business Process Execution Language)
- SAML (Security Assertion ML)
- UBL (Universal Business Language)

BuilConn™

Phase IV: Abstraction

- WSDM-MOWS (Management of Web Services)
- Internationalization
- OWL – Ontology Web Language
  - OWL is designed for use by applications that need to process the content of information instead of just presenting information to humans. OWL facilitates greater machine interpretability of Web content than that supported by XML, RDF, and RDF Schema (RDF-S) by providing additional vocabulary along with a formal semantics. OWL is a W3C specification

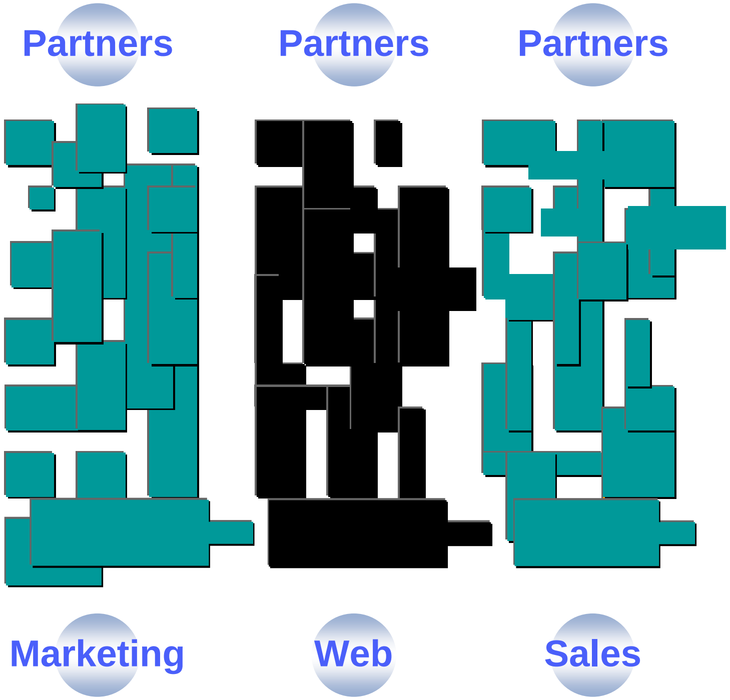
building the Semantic Web...

OASIS

Better integration

"40% of IT spending is on integration"
— IDC

" Every $1 for software = $7 to $9 on integration"
— Gartner

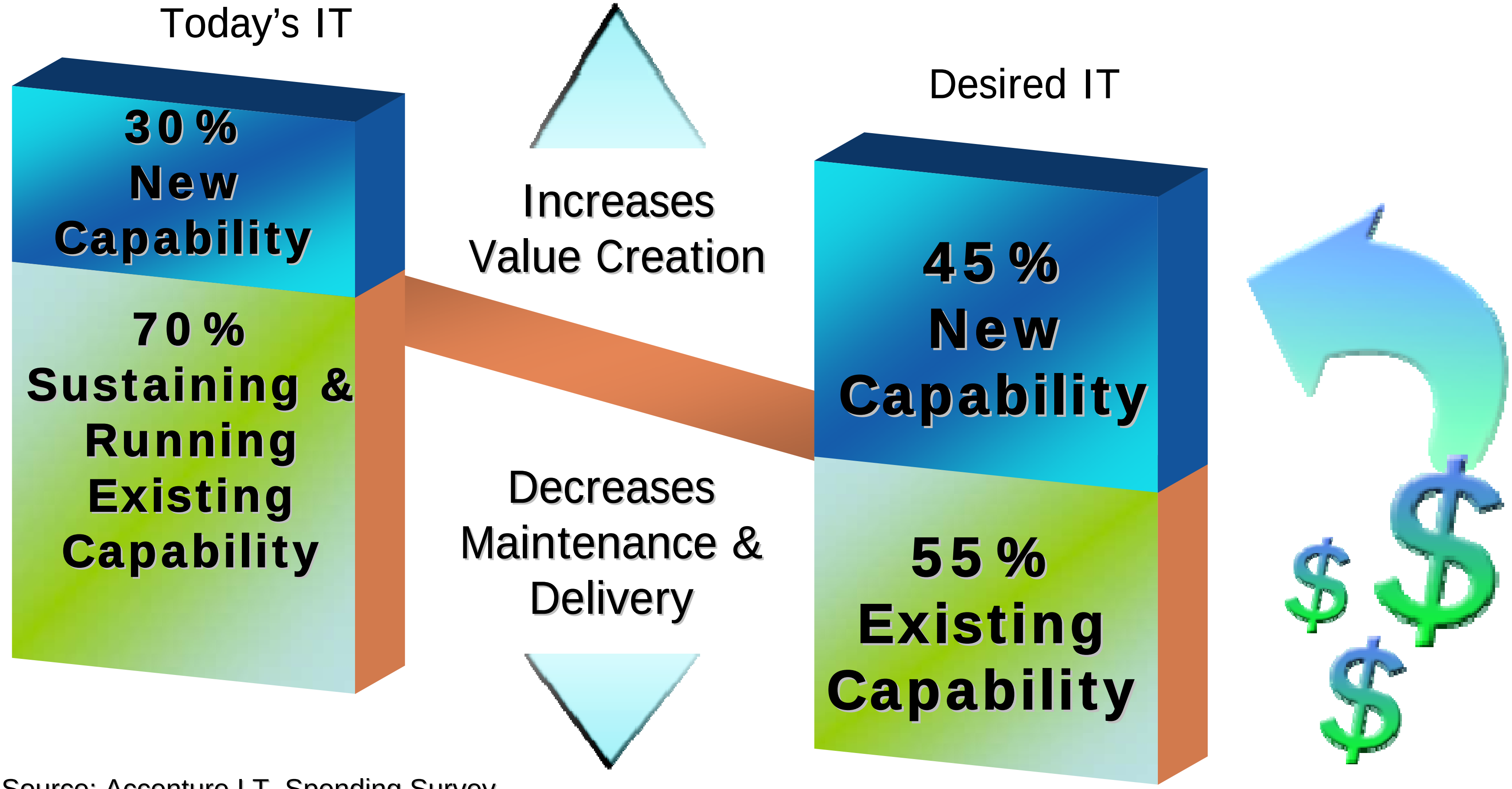


Historical limitations:

- Monolithic applications can't be reused
- Ad hoc integration creates connections that are difficult to change/maintain
- Lack of standards limits ability to deliver meaningful interoperability

BuilConn™

Companies want IT to deliver more business value

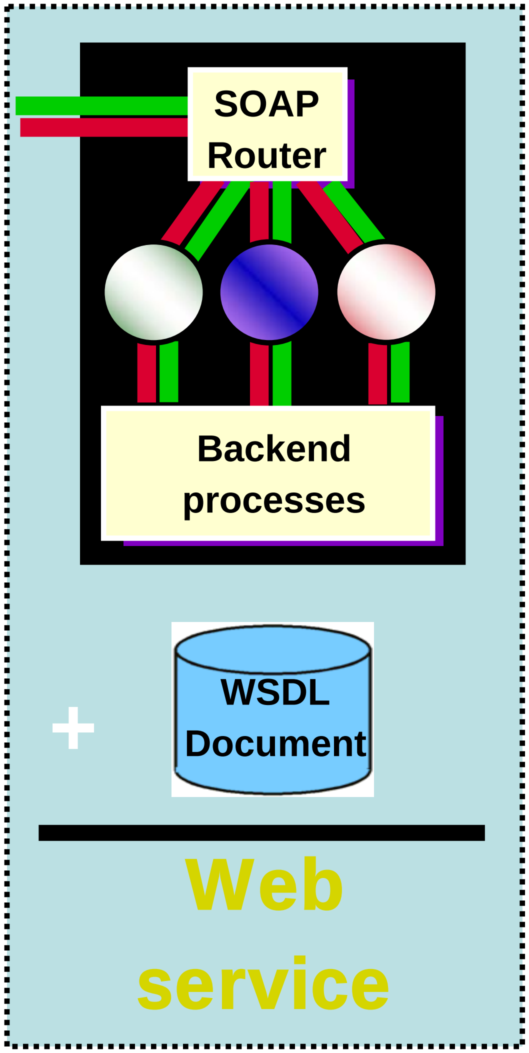


Source: Accenture I.T. Spending Survey

OASIS

What is a Web service?

- A Web service is:
  - a **software component** whose interface is **described via WSDL**
  - is capable of being accessed via **standard** network protocols such as SOAP over HTTP.
  - a software system designed to support interoperable machine-to-machine interaction over a network.
  - easy to combine and recombine to meet the needs of customers, suppliers and business partners because it is:
    - built on open standards and therefore do not require custom-coded connections for integration
    - self-contained and modular



What is SOA?

- ***A service-oriented architecture (SOA)*** is an enterprise-scale IT system architecture in which application functions are built as business aligned components (or "services") that are loosely-coupled and well-defined to support interoperability, and to improve flexibility and re-use.
  - An SOA separates out the concerns of the Service requestors and Service Providers (and Brokers).

- **A Service** is a discoverable software resource which has a service description. The service description is available for searching, binding and invocation by a service requestor. The service description implementation is realized through a service provider who delivers quality of service requirements for the service requestor. Services can be governed by declarative policies.
  - SOA is not a product – it is about aligning IT and business needs

BuilConn™

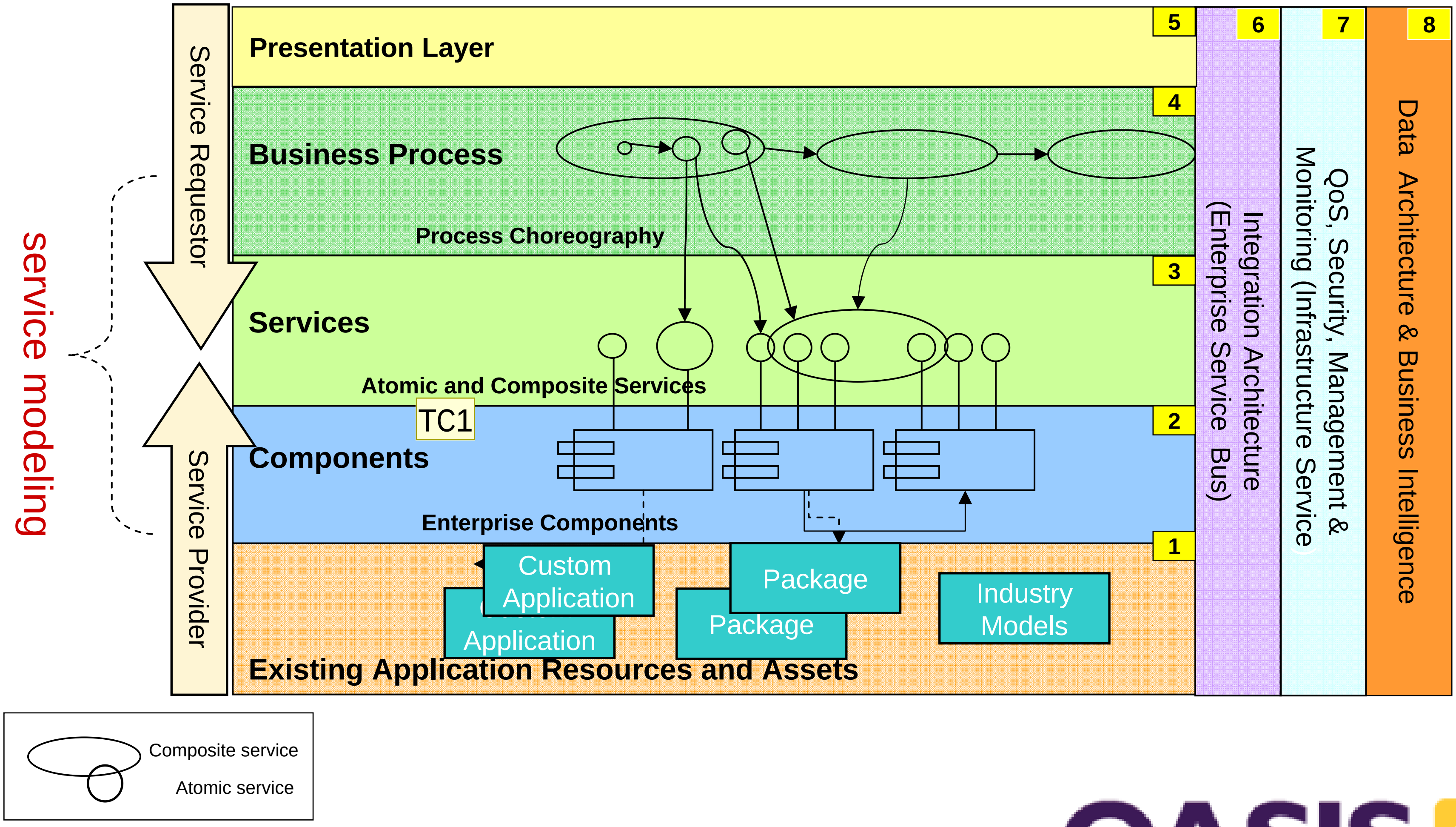
An IT Consultant view of Web Services

- Web services can be a part of the answer
- Service Oriented Architecture (SOA) is another part
- The two are not the same thing:
  - Most of today's production Web services systems aren't service oriented architectures - they're simple remote procedure calls or point-to-point messaging via SOAP or well structured integration architectures
  - Most of today's production service oriented architectures don't primarily use Web services - they use ftp, batch files, asynchronous messaging etc. - mature technologies
- Achieving the promoted benefits requires *both* SOA and Web services
- Organizations should get interested in the *combination* of SOA + Web services
  - business flexibility requires IT flexibility
  - business flexibility enables a company to support the one constant of change business

Thanks to Steve Graham, whose PowerPoints I stole.

OASIS

Layered SOA



Presentation Layer
5
Business Process
Process Choreography
4
Services
Atomic and Composite Services
3
TC1
Components
Enterprise Components
2
Custom Application
Custom Application
Package
Package
Industry Models
Existing Application Resources and Assets
1
Integration Architecture (Enterprise Service Bus)
6
QoS, Security, Management & Monitoring (Infrastructure Service)
7
Data Architecture & Business Intelligence
8
Service Requestor
Service Provider
service modeling
Composite service
Atomic service

Slide 15

TC1 Bob:

oBIX is struggling with level 2 - we need to be on level 3 to be managed by 4

Toby Considine, 15-Feb-06

How do Enterprise Standards Grow? Phase I

- Small, tight specifications
  - Fully functional
  - Limited Interoperability
  - Easy to implement

BuilConn™

How do Standards Grow?
Phase II

- Component Sockets
  - Moving from Process to Service
    - Abstraction
    - Profile
    - Domain-Specific Language
    - Component
- Conformance Testing required
  - Or interoperability will be impossible

OASIS

oBIX as an Enterprise Web Service

- oBIX is about Buildings and the Enterprise based upon Enterprise IT standards and Enterprise architectures

- oBIX is at the earliest phase of development as defined above

oBIX as Web Service to date

- Small light protocol
- Optimized for easy implementation
- Standard functions common to all control systems

Characteristics of oBIX Today

- REST works the way current control systems work, and so offers an easy transition to existing controls integrators.
  - REST also allows easy development of AJAX-style interfaces, offering immediate benefits in upgrading deliverable interfaces to the early adopter.
  - REST provides the best platform for the immediate implementation of monitoring and control functions.
- Deeper integration with enterprise systems will require SOAP.
  - Such integration will also require componentized abstractions, or profiles, which can and now will, be developed on the small tight v1.0 platform.
- *By supporting both SOAP and REST, oBIX 1.0 allows rapid (and easy) benefits for early adopters while supporting the incremental extension and componentization that long-term enterprise integration will require.*

Moving oBIX up to the Enterprise

- Use Web services and SOA to make IT systems and Building Automation ***easier to integrate***
- Define common profiles and services based upon core protocol
- Define compliance suites

oBIX Mid-Term Goals

- Evolve to support composite frameworks
- Re-use related Namespaces
  - UnitsML starting as OASIS TC
- Provides an abstraction over base Building Automation data
- Get building automation systems “on the [enterprise] bus”

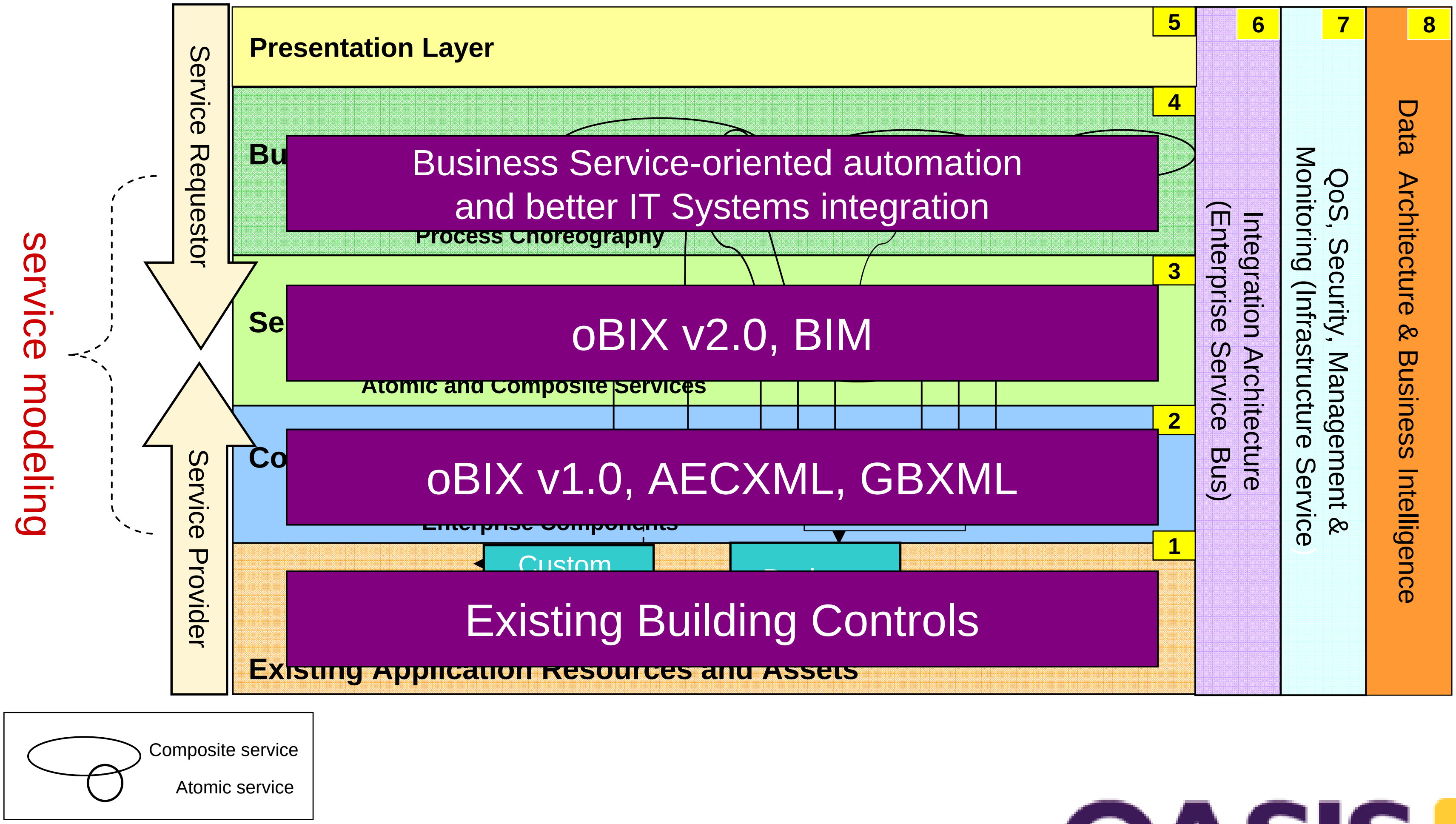
BuilConn™

oBIX Mid-Term Goals

- Full Participant in NBIM
  - Support of COBIE
- Transforms to GBXML and "Continuous Commissioning"
- Support of Emergency Response
  - CAP and EDXL compatibility
  - OGC Interoperability
    - Open Geospatial Consortium

OASIS

Layered *Building Automation* SOA Standards


service modeling
Service Requestor
Service Provider
Presentation Layer
5
4
3
2
1
6
7
8
Bu
Business Service-oriented automation and better IT Systems integration
Process Choreography
Se
oBIX v2.0, BIM
Atomic and Composite Services
Co
oBIX v1.0, AECXML, GBXML
Enterprise Components
Custom
Existing Building Controls
Existing Application Resources and Assets
Integration Architecture (Enterprise Service Bus)
QoS, Security, Management & Monitoring (Infrastructure Service)
Data Architecture & Business Intelligence
Composite service
Atomic service

BuilConn™

We need work on the Enterprise Abstraction Models

- Capabilities Models
  - One for each control silo
- Analytics Models
  - M&V – Self Commissioning
- Tenant Models
  - Cross-Silos “The Room”
  - Uses: Schedules and Variable Costs

Can you help?

OASIS